

CLAIMS

1. 1. Apparatus for representing and managing an XML-compliant document in a
2. memory, the XML-compliant document being composed of a plurality of elements
3. arranged in a nested relationship, the apparatus comprising:
 4. a data document including a plurality of element objects, each element
5. object representing a part of the XML-compliant document; and
 6. a mechanism for arranging the plurality of element objects in a hierarchy
7. representative of the nested relationship of the elements.
1. 2. Apparatus as recited in claim 1 wherein at least some of the elements contain
2. textual content and wherein element objects representing the elements contain
3. the textual content.
1. 3. Apparatus as recited in claim 1 wherein at least some of the elements contain
2. attributes having values and wherein element objects representing the elements
3. contain the attribute values.
1. 4. Apparatus as recited in claim 3 wherein the attribute values contained in the at
2. least some elements are typed.
1. 5. Apparatus as recited in claim 3 further comprising an attribute index containing
2. consistent pointers to all element objects containing attribute values.
1. 6. Apparatus as recited in claim 1 wherein the arranging mechanism comprises
2. database pointers and wherein a database pointer in a parent element object
3. points to child objects of the parent element object in order to arrange the parent
4. object and child objects in a hierarchical relationship.

- 1 7. Apparatus as recited in claim 1 further comprising a schema document
- 2 referenced by the data document, the schema document containing content that
- 3 describes the pattern of element objects and attributes, the existence and
- 4 structure of document indicies, and commonly used strings in the data document.
- 1 8. Apparatus as recited in claim 7 wherein the schema document is referenced by
- 2 an XML processing statement in the data document.
- 1 9. Apparatus as recited in claim 1 further comprising a binary document object for
- 2 representing a data document containing binary data.
- 1 10. Apparatus as recited in claim 1 further comprising a document object for
- 2 representing the data document.
- 1 11. Apparatus as recited in claim 10 wherein the document object contains links to
- 2 other document objects so that the other document objects are sub-documents of
- 3 the document object.
- 1 12. Apparatus as recited in claim 1 wherein each of the element objects exports a
- 2 uniform interface containing methods for manipulating each of the element
- 3 objects.
- 1 13. Apparatus for binding program code to portions of an XML-compliant document
- 2 composed of a plurality of elements, each of which is identified by a tag, the
- 3 elements being arranged in a nested relationship, the apparatus comprising:
 - 4 a data document including a plurality of element objects, each element
 - 5 object representing a part of the XML-compliant document, the plurality of
 - 6 element objects being arranged in a hierarchy representative of the nested
 - 7 relationship of the elements;

8 a schema document referenced by the data document, the schema
9 document containing a registry which maps a tag identifying one of the elements
10 to a program ID code; and

11 a mechanism that uses the program ID code to construct an object
12 containing the program code.

1 14. Apparatus as recited in claim 13 wherein the registry is a two-column table that
2 maps element tags to program ID codes.

1 15. Apparatus as recited in claim 13 wherein the mechanism is responsive to a
2 method call for retrieving the program ID code for constructing the object
3 containing the program code.

1 16. Apparatus as recited in claim 13 wherein the mechanism is the COM object
2 manager and the program ID code is a ProgID code.

1 17. Apparatus as recited in claim 16 wherein the COM manager comprises a locating
2 mechanism that uses the ProgID code to locate the program code and an object
3 constructor that constructs an object incorporating the located program code.

1 18. Apparatus as recited in claim 13 wherein the schema document is referenced in
2 the data document by an XML processing statement.

1 19. Apparatus for representing and managing an XML-compliant document in a
2 memory, the XML-compliant document being updated concurrently by a first
3 process having a first address space in the memory and second process having
4 a second address space in the memory, the apparatus comprising:

5 a first storage manager controlled by the first process that constructs, from
6 class code in the first address space, at least one document object including first

7 data representing a part of the XML-compliant document stored in the first
8 address space;

9 a second storage manager controlled by the second process that
10 constructs, from class code in the second address space which class code is
11 identical to the class code in the first address space, at least one document
12 object including second data representing a part of the XML-compliant document
13 stored in the second address space; and
14 a synchronization mechanism that insures that the first data and the
15 second data are continually equated.

1 20. Apparatus as recited in claim 19 wherein the first data is stored in a region
2 mapped into the first address space and the second data is stored in the same
3 region mapped into the second address space and the synchronization
4 mechanism continually equates the region data mapped in the first and second
5 address spaces.

6 21. Apparatus as recited in claim 20 wherein the second process comprises a
7 mechanism for requesting a copy of the region data from the first address space
8 if the second address space does not have the most recent copy of the region
9 data.

10 22. Apparatus as recited in claim 20 wherein the first process comprises methods for
11 requesting that the synchronization manager lock the region data when the first
12 process is changing the region data in the first address space.

13 23. Apparatus as recited in claim 20 wherein the second process comprises methods
14 for requesting that the synchronization manager lock the region data when the
15 second process is changing the region data in the second address space.

1 24. Apparatus as recited in claim 20 wherein the first process can perform read and
2 write operations on the region and wherein the apparatus further comprises a
3 mechanism for grouping a plurality of the read and write operations into a
4 transaction.

1 25. Apparatus as recited in claim 24 wherein the first process comprises methods for
2 requesting that the synchronization manager lock the region data during the
3 processing of all read and write operations in a transaction.

1 26. Apparatus as recited in claim 25 further comprising a logging system that
2 periodically writes recovery log entries to a persistent database during the
3 processing of all read and write operations in a transaction.

1 27. Apparatus as recited in claim 19 wherein the first process comprises a storage
2 mechanism for storing a copy of the region data in a non-volatile store.

1 28. Apparatus as recited in claim 27 wherein the non-volatile store comprises an
2 object store.

1 29. Apparatus as recited in claim 27 wherein the non-volatile store comprises a file
2 system.

1 30. Apparatus as recited in claim 19 wherein the synchronization mechanism
2 comprises a distributed memory system.

1 31. Apparatus as recited in claim 19 wherein both the first and second address
2 spaces contain equivalent program code for manipulating the first and second
3 document objects.

1 32. Apparatus as recited in claim 19 wherein the first and second storage manager
2 each construct a cross-process synchronization object that is used to
3 synchronize the first and second processes.

1 33. Apparatus for representing and managing an XML-compliant document in a
2 memory, the XML-compliant document being composed of a plurality of
3 elements, the elements being arranged in a nested relationship, the apparatus
4 comprising:
5 a data document including a plurality of element objects, each element
6 object representing a part of the XML-compliant document; and
7 a collection manager that maps the element objects into a tabular data
8 structure including an index structure.

1 34. Apparatus as recited in claim 33 further comprising a record set engine that is
2 responsive to user commands for propagating a set of updates for the tabular
3 data structure to the collection manager.

1 35. Apparatus as recited in claim 34 wherein the collection manager further
2 comprises an update mechanism which responds the set of updates by updating
3 the index structure.

1 36. Apparatus as recited in claim 35 wherein the collection manager further
2 comprises a notification system that notifies the users when changes are made
3 to the tabular data structure.

1 37. Apparatus as recited in claim 36 wherein the collection manager further
2 comprises a navigation mechanism for creating a cursor to allow the users to
3 navigate within the tabular data structure.

1 38. A method for representing and managing an XML-compliant document in a
2 memory, the XML-compliant document being composed of a plurality of elements
3 arranged in a nested relationship, the method comprising:
4 (a) creating a data document in the memory including a plurality of element
5 objects, each element object representing a part of the XML-compliant
6 document; and
7 (b) arranging the plurality of element objects in a hierarchy representative of
8 the nested relationship of the elements.

1 39. A method as recited in claim 38 wherein at least some of the elements contain
2 textual content and wherein element objects representing the elements contain
3 the textual content.

40. A method as recited in claim 38 wherein at least some of the elements contain
4 attributes having values and wherein element objects representing the elements
5 contain the attribute values.

41. A method as recited in claim 40 wherein the attribute values contained in the at
4 least some elements are typed.

42. A method as recited in claim 40 further comprising an attribute index containing
2 consistent pointers to all element objects containing attribute values.

43. A method as recited in claim 38 wherein step (b) comprises creating a database
2 pointer in a parent element object which pointer points to child objects of the
3 parent element object in order to arrange the parent object and child objects in a
4 hierarchical relationship.

1 44. A method as recited in claim 38 further comprising (c) creating a schema
2 document referenced by the data document in the memory, the schema
3 document containing content that describes the pattern of element objects and
4 attributes, the existence and structure of document indicies, and commonly used
5 strings in the data document.

1 45. A method as recited in claim 44 wherein step (c) comprises creating the schema
2 document referenced by an XML processing statement in the data document.

1 46. A method as recited in claim 38 further comprising (d) creating a binary
2 document object in the memory for representing a data document containing
3 binary data.

1 47. A method as recited in claim 38 further comprising (e) creating a document
2 object in the memory for representing the data document.

1 48. A method as recited in claim 47 wherein the document object contains links to
2 other document objects so that the other document objects are sub-documents of
3 the document object.

1 49. A method as recited in claim 38 wherein each of the element objects exports a
2 uniform interface containing methods for manipulating each of the element
3 objects.

1 50. A method for binding program code in a memory to portions of an XML-compliant
2 document composed of a plurality of elements, each of which is identified by a
3 tag, the elements being arranged in a nested relationship, the XML-compliant
4 document being stored in the memory and the method comprising:
5 (a) creating a data document in the memory including a plurality of element
6 objects, each element object representing a part of the XML-compliant

7 document, the plurality of element objects being arranged in a hierarchy
8 representative of the nested relationship of the elements;
9 (b) creating a schema document referenced by the data document in the
10 memory, the schema document containing a registry which maps a tag
11 identifying one of the elements to a program ID code; and
12 (c) using the program ID code to construct an object containing the program
13 code.

1 51. A method as recited in claim 50 wherein the registry is a two-column table that
2 maps element tags to program ID codes.

1 52. A method as recited in claim 50 wherein step (c) is initiated in response to a
2 method call for retrieving the program ID code.

1 53. A method as recited in claim 50 wherein step (c) is performed by the COM object
2 manager and the program ID code is a ProgID code.

1 54. A method as recited in claim 53 wherein the COM manager performs the steps of
2 using the ProgID code to locate the program code and constructing an object
3 incorporating the located program code.

1 55. A method as recited in claim 50 wherein the schema document is referenced in
2 the data document by an XML processing statement.

1 56. A method for representing and managing an XML-compliant document in a
2 memory, the XML-compliant document being updated concurrently by a first
3 process having a first address space in the memory and second process having
4 a second address space in the memory, the method comprising:

5 (a) using a first storage manager controlled by the first process to construct,
6 from class code in the first address space, at least one document object

7 including first data representing a part of the XML-compliant document
8 stored in the first address space;
9 (b) using a second storage manager controlled by the second process to
10 construct, from class code in the second address space which class code
11 is identical to the class code in the first address space, at least one
12 document object including second data representing a part of the XML-
13 compliant document stored in the second address space; and
14 (c) insuring that the first data and the second data are continually equated.

1 57. A method as recited in claim 56 wherein the first data is stored in a region
2 mapped into the first address space and the second data is stored in the same
3 region mapped into the second address space and step (c) comprises continually
54 equating the region data mapped in the first and second address spaces.
55
56
57
58. A method as recited in claim 57 wherein step (c) comprises requesting a copy of
59 the region data from the first address space if the second address space does
60 not have the most recent copy of the region data.
61
62
63
64
65
66
67
68
69
70
71 59. A method as recited in claim 57 wherein step (c) comprises locking the region
72 data when the first process is changing the region data in the first address space.
73
74
75
76
77
78
79
80
81 60. A method as recited in claim 57 wherein step (c) comprises locking the region
82 data when the second process is changing the region data in the second address
83 space.
84
85
86
87
88
89
90
91 61. *A method*
92 *Apparatus* as recited in claim 57 wherein the first process can perform read and
93 write operations on the region and wherein the method further comprises (d)
94 grouping a plurality of the read and write operations into a transaction.

KGM
RLR
ESF
R.O.

1 62. A method as recited in claim 61 wherein step (c) comprises locking the region
2 data during the processing of all read and write operations in a transaction.

1 63. A method as recited in claim 62 wherein step (c) further comprises periodically
2 writing recovery log entries to a persistent database during the processing of all
3 read and write operations in a transaction.

1 64. A method as recited in claim 56 further comprising (e) under the control of the
2 first process, storing a copy of the region data in a non-volatile store.

1 65. A method as recited in claim 64 wherein the non-volatile store comprises an
2 object store.

61
62
63
64
65
66. A method as recited in claim 64 wherein the non-volatile store comprises a file
system.

67. A method as recited in claim 56 wherein step (c) is performed by a distributed
memory system.

68. A method as recited in claim 56 further comprising (f) manipulating the first and
second document objects with equivalent program code in both the first and
second address spaces.

69. A method as recited in claim 56 further comprising (g) constructing a cross-
process synchronization object that is used to synchronize the first and second
processes.

70. A method for representing and managing an XML-compliant document in a
memory, the XML-compliant document being composed of a plurality of

3 elements, the elements being arranged in a nested relationship, the method
4 comprising:

5 (a) creating a data document in the memory including a plurality of element
6 objects, each element object representing a part of the XML-compliant
7 document; and
8 (b) mapping the element objects into a tabular data structure including an
9 index structure.

1 71. A method as recited in claim 70 further comprising (c) propagating a set of
2 updates for the tabular data structure to the collection manager in response to
3 user commands.

51 72. A method as recited in claim 71 further comprising (d) updating the index
52 structure in response to set of updates.

53 73. A method as recited in claim 72 further comprising (e) notifying the users when
54 changes are made to the tabular data structure.

55 74. A method as recited in claim 73 further comprising (f) creating a cursor to allow
56 the users to navigate within the tabular data structure.

57 75. A computer program product for representing and managing an XML-compliant
58 document in a memory, the XML-compliant document being composed of a
59 plurality of elements arranged in a nested relationship, the computer program
60 product comprising a computer usable medium having computer readable
61 program code thereon, including:

62 program code for creating a data document in the memory including a
63 plurality of element objects, each element object representing a part of the XML-
64 compliant document; and

program code for arranging the plurality of element objects in a hierarchy representative of the nested relationship of the elements.

76. A computer program product for binding program code in a memory to portions of an XML-compliant document composed of a plurality of elements, each of which is identified by a tag, the elements being arranged in a nested relationship, the XML-compliant document being stored in the memory and the computer program product comprising a computer usable medium having computer readable program code thereon, including:

program code for creating a data document in the memory including a plurality of element objects, each element object representing a part of the XML-compliant document, the plurality of element objects being arranged in a hierarchy representative of the nested relationship of the elements;

program code for creating a schema document referenced by the data document in the memory, the schema document containing a registry which maps a tag identifying one of the elements to a program ID code; and

program code for using the program ID code to construct an object containing the program code.

77. A computer program product for representing and managing an XML-compliant document in a memory, the XML-compliant document being updated concurrently by a first process having a first address space in the memory and second process having a second address space in the memory, the computer program product comprising a computer usable medium having computer readable program code thereon, including:

program code for using a first storage manager controlled by the first process to construct, from class code in the first address space, at least one document object including first data representing a part of the XML-compliant document stored in the first address space;

11 program code for using a second storage manager controlled by the
12 second process to construct, from class code in the second address space which
13 class code is identical to the class code in the first address space, at least one
14 document object including second data representing a part of the XML-compliant
15 document stored in the second address space; and

16 program code for insuring that the first data and the second data are
17 continually equated.

1 78. A computer program product for representing and managing an XML-compliant
2 document in a memory, the XML-compliant document being composed of a
3 plurality of elements, the elements being arranged in a nested relationship, the
4 computer program product comprising a computer usable medium having
5 computer readable program code thereon, including:

6 program code for creating a data document in the memory including a
7 plurality of element objects, each element object representing a part of the XML-
8 compliant document; and

9 program code for mapping the element objects into a tabular data
10 structure including an index structure.

11 79. A computer data signal embodied in a carrier wave for representing and
12 managing an XML-compliant document in a memory, the XML-compliant
13 document being composed of a plurality of elements arranged in a nested
14 relationship, the computer data signal comprising:

15 program code for creating a data document in the memory including a
16 plurality of element objects, each element object representing a part of the XML-
17 compliant document; and

18 program code for arranging the plurality of element objects in a hierarchy
19 representative of the nested relationship of the elements.

1 80. A computer data signal embodied in a carrier wave for binding program code in a
2 memory to portions of an XML-compliant document composed of a plurality of
3 elements, each of which is identified by a tag, the elements being arranged in a
4 nested relationship, the XML-compliant document being stored in the memory
5 and the computer data signal comprising:
6 program code for creating a data document in the memory including a
7 plurality of element objects, each element object representing a part of the XML-
8 compliant document, the plurality of element objects being arranged in a
9 hierarchy representative of the nested relationship of the elements;
10 program code for creating a schema document referenced by the data
11 document in the memory, the schema document containing a registry which
12 maps a tag identifying one of the elements to a program ID code; and
13 program code for using the program ID code to construct an object
14 containing the program code.

1 81. A computer data signal embodied in a carrier wave for representing and
2 managing an XML-compliant document in a memory, the XML-compliant
3 document being updated concurrently by a first process having a first address
4 space in the memory and second process having a second address space in the
5 memory, the computer data signal comprising:
6 program code for using a first storage manager controlled by the first
7 process to construct, from class code in the first address space, at least one
8 document object including first data representing a part of the XML-compliant
9 document stored in the first address space;
10 program code for using a second storage manager controlled by the
11 second process to construct, from class code in the second address space which
12 class code is identical to the class code in the first address space, at least one
13 document object including second data representing a part of the XML-compliant
14 document stored in the second address space; and

15 program code for insuring that the first data and the second data are
16 continually equated.

1 82. A computer data signal for representing and managing an XML-compliant
2 document in a memory, the XML-compliant document being composed of a
3 plurality of elements, the elements being arranged in a nested relationship, the
4 computer data signal comprising:
5 program code for creating a data document in the memory including a
6 plurality of element objects, each element object representing a part of the XML
7 compliant document; and
8 program code for mapping the element objects into a tabular data
9 structure including an index structure.